IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR ELECTRIC POWER SYSTEMS

Received May 26, 2019, accepted June 10, 2019, date of publication June 20, 2019, date of current version July 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924030

Demand Response Management for Industrial Facilities: A Deep Reinforcement Learning Approach

XUEFEI HUANG[1], SEUNG HO HONG[1,2], (Senior Member, IEEE), MENGMENG YU[1], (Member, IEEE), YUEMIN DING[2], (Member, IEEE), AND JUNHUI JIANG[1]

[1]Department of Electronic Engineering, Hanyang University, Ansan 15588, South Korea
[2]School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300000, China

Corresponding author: Seung Ho Hong (shhong@hanyang.ac.kr)

This work was supported in part by the National Research Foundation of Korea under the Framework of International Cooperation Program, South Korea–China, under Grant NRF-2016K2A9A2A11938310, and in part by the National Research Foundation of Korea under Grant NRF-2018K1A3A1A61026320.

ABSTRACT As a major consumer of energy, the industrial sector must assume the responsibility for improving energy efficiency and reducing carbon emissions. However, most existing studies on industrial energy management are suffering from modeling complex industrial processes. To address this issue, a model-free demand response (DR) scheme for industrial facilities was developed. In practical terms, we first formulated the Markov decision process (MDP) for industrial DR, which presents the composition of the state, action, and reward function in detail. Then, we designed an actor-critic-based deep reinforcement learning algorithm to determine the optimal energy management policy, where both the actor (Policy) and the critic (Value function) are implemented by the deep neural network. We then confirmed the validity of our scheme by applying it to a real-world industry. Our algorithm identified an optimal energy consumption schedule, reducing energy costs without compromising production.

INDEX TERMS Artificial intelligence, deep reinforcement learning, demand response (DR), industrial facilities, actor-critic.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Given the continuous growth of the global population and the increasing demand for energy, sustainable and efficient utilization of power resources has always been a high priority to avoid an energy crisis [1]. The industrial sector has long consumed over 50% of global energy [2], and the energy-related CO_2 emissions from this sector are predicted to grow to an extent greater than emissions from other sectors between 2017 and 2050 [1]. A well-designed energy management scheme for industrial facilities would alleviate pressure on the power grid, improving energy efficiency and cutting carbon emissions. To achieve this target, demand response (DR) is a promising approach that motivates end-users with flexible loads to vary their energy consumption in response to dynamic electricity prices or other incentives [3]. In recent years, the effect of DR on energy savings has attracted a great deal of attention [4], [5]; moreover, DR is also regarded as one of the key drivers of progress in smart grid technology [6].

However, compared to the large degree of DR participation in the residential and commercial sectors, the industrial potential of DR is not well understood [7]. Over the past decade, the Lawrence Berkeley National Laboratory carried out a long-term program to tap into the potential of DR for different industrial areas, includingwastewater treatment plants, agricultural irrigation, refrigerated warehouses, and so on [8]. They identified barriers to widespread application of DR in industry, from the following two viewpoints [9]. The first problem is industrial diversity. Unlike the residential and commercial sectors, the energy consumption of production lines using diverse items of equipment varies considerably. Successful DR implementation requires a high-resolution

The associate editor coordinating the review of this manuscript and approving it for publication was Canbing Li.

model capturing the physical characteristics of all equipment in the system [7]. However, it is not easy to model an entire industrial facility. Specifically, apart from electricity management, raw, intermediate, and auxiliary resources for production must all be considered. The second problem is the need to guarantee daily production; potentially, DR could result in production losses or cost increases because production must be shifted [9]. Thus, industrial customers are wary of DR programs. To overcome these, there is a pressing need for a solution that avoids complex modeling while maintaining production and minimizing energy costs; an artificial intelligence-based deep reinforcement learning (RL) method can be used to this end.

Deep RL [10], [11] is a combination of RL [12] and deep learning [13]. RL is derived from the psychological theory of the same name, and develops policies that stochastically optimize control strategies by employing trajectories produced by interactions with a real system [14]. In practical terms, deep RL has found applications in robotics [15], financial trading [16], allocation of communication resources [17], and vehicular networking [18]. In certain tests, deep RL outperforms human experts in terms of optimal decision-making; a famous example is the AlphaGo [19].

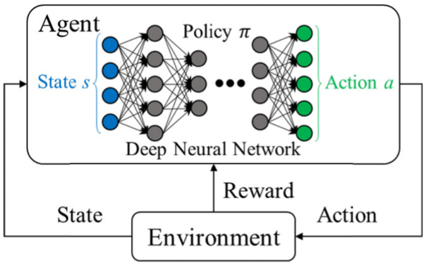


FIGURE 1. Schematic structure of deep reinforcement learning agent.

Unlike traditional methods that employ dynamic programming techniques [20], RL does not require a detailed mathematical model of the system to ensure optimal control. Rather, the RL agent regards the target system as its environment, and then optimizes the control policy by interacting with it. More specifically, the agent/environment interaction proceeds in discrete steps. During each step, depending on the current state of the environment, the agent chooses an action based on its current policy. The environment feeds back a reward and then enters the next state. Thus, the RL agent learns to adjust its policy by reference to the relationships among the state, action, and reward. After gaining sufficient experience, the RL agent can determine an optimal policy associated with the maximum cumulative reward. For example, Figure 1 shows a schematic of a deep RL agent. Within the agent/environment interaction of the RL, the deep neural network maintains the internal policy of the agent, which determines the next action based on the current state of the environment.

B. LITERATURE REVIEW

Taking advantage of its model-free characteristics, RL has been applied in several studies for solving energy management problems, which can be identified into four major groups [21]: heating ventilation and air conditioning (HVAC) systems, electric vehicles (EVs), smart home appliances, and distributed generation with energy storage. HVAC is the common area for providing DR capabilities, which can contribute to load curtailment events by modifying the temperature set points, participating in load shifting according to price dynamic [22]. For example, in [23]–[25], Ruelens *et al.* performed batch RL-based studies involving electric water heaters and heat pump thermostats; determining an optimal temperature control policy (i.e., that with the lowest energy cost). In [26], De Somer *et al.* demonstrated a data-driven control approach for DR in residential buildings, using RL to optimally schedule the heating cycles of domestic hot water buffers to maximize the self-consumption of local photovoltaic production. In [27], Patyn *et al.* discussed the implementation of RL for heat pumps in a DR setting and its cost-effectiveness in comparison to different types of neural networks. The increasing utilization of EVs holds a great DR potential by their electrical storage capacity as well as their inherent connectivity [21]. In [28], Wan *et al.* developed an RL-based energy management scheme for electric vehicles, in which the scheduling problem is formulated as a Markov decision process (MDP). A model-free approach based on deep reinforcement learning was proposed to identify the optimal charging strategy. Also, in work with electric vehicles, Chiş *et al.* [29] developed an RL-based algorithm to solve a DR problem: how much daily energy should be added to plug-in electric vehicle batteries? Vandael *et al.* [30] used a batch RL method to develop a day-ahead charging plan for an electric vehicle fleet; plug-in time, power limitations, battery size, and power curves were all considered. In 2010, O'Neill *et al.* [31] introduced the application of RL to control smart home appliances. In the same area, Wen *et al.* [32] developed a device-based DR scheme for residential and small commercial buildings; the energy scheduling problem was decomposed into several device clusters and reformulated as an RL problem. Kaliappan *et al.* [33] used a RL algorithm to control a set of home appliances, which considered the balance between energy consumption and discomfort of the consumer. Liu *et al.* [34] focused on reducing the electricity cost of shiftable loads. In the application related to distributed generation with energy storage. Qiu *et al.* [35] used RL to minimize energy consumption by controlling a battery under the presence of solar PV panels. Kofinas *et al.* [36] proposed a decentralized cooperative multi-agent RL algorithm to maximize the self-energy consumption in a microgrid.

However, although several RL-based energy management algorithms are available, few can be applied in industry, for two reasons. First, most feature relatively simple implementation scenarios, considering only the operation of individual items of equipment, such as those amenable to thermostatically controlled loading, and electric vehicles; interactions among different types of equipment are not considered. However, in a real-world production line, almost every task requires different equipment. Second, most studies considered only electricity costs; they thus sought to minimize energy costs only. However, although reducing energy costs is desirable considering the overall operations costs of an industrial facility, normal production cannot be compromised.

Considering all of the factors mentioned above, in this paper we develop a deep RL-based DR scheme for industrial facilities. First, we introduce a MDP framework suited for industrial DR; this considers not only energy but also production resource management. Moreover, the designed reward function does not compromise production but still minimizes energy costs. Next, based on this framework, we design an RL algorithm that optimizes energy management. Specifically, we formulate a variant of the actor-critic algorithm [37], [38] that is suitable for hourly price-based DR to accelerate the learning process. Moreover, as many equipment items and resources must be considered simultaneously, we use the deep neural network [39], which acts as the function approximator, to map the relationships among the state, action, and reward. Finally, the proposed scheme is validated by applying it to steel powder manufacturing (SPM), i.e., a real-world industry.

To the best of our knowledge, this is the first proposal of using deep RL to manage the DR of industrial facilities characterized by the simultaneous operation of many types of equipment associated with the need for various resources (i.e., electricity and production materials). The remainder of the paper is organized as follows. The problem is formulated in Section II, which introduces industrial energy management and the corresponding MDP framework. Section III describes our deep RL algorithm. Section IV is a real-world manufacturing case study showing how our scheme can be implemented. Section V analyzes the simulation results. Section VI provides the conclusion and future work.

II. PROBLEM FORMULATION

Here, we first introduce industrial energy management and then present our MDP framework.

A. SYSTEM DESCRIPTION

1) GENERAL ARCHITECTURE

In accordance with the standard of [40], Figure 2 shows an energy management model for industrial facilities, illustrating the interrelationships of various elements. In detail, these elements include a smart grid (SG), smart meter (SM), gateway (G), production planner, factory energy management system (FEMS), energy load, utility power line (UPL), facility power line (FPL), wide area network (WAN), and local area network (LAN). From the macro perspective, the system can be divided into two parts; the SG belongs to the utility side and the industrial facility belongs to the demand side. Between them, the smart meter and gateway serve as the interface for energy delivery and information exchange. More specifically, the smart meter records electricity transmission between the UPL and FPL. The gateway transmits messages (such as electricity prices) between the WAN and LAN. On the demand side, the LAN enables message exchange among elements, and the FPL distributes the electricity. The FEMS serves as the system core, controlling all energy management. In detail, during operation, depending on the target set by the production planner and the electricity price notified by the gateway, the FEMS determines a working schedule for every energy load based on pre-installed energy management algorithms and strategies. The energy load is that of all energy consumers. These can be divided into three types [40]: non-shiftable equipment (NSE), shiftable equipment (SE), and controllable equipment (CE). Specifically, the energy demands of NSE must always be satisfied, independent of the electricity price. The energy demands of SE and CE are adjustable; SE can be switched ON or OFF and the energy demand of CE is controllable, exploiting various pre-specified multiple operating points (OPs). Therefore, the energy demand of equipment with only a single OP (i.e., NSE) is fixed, whereas the demands of those with multiple OPs (i.e., CE and SE) are flexible depending on the working state; thus, SE and CE are prime candidates for industrial DR.

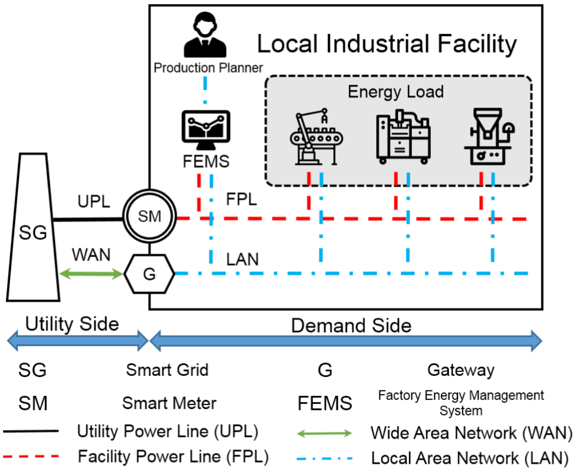


FIGURE 2. Energy management model for industrial facility.

As mentioned in Section I, in addition to power usage, production materials must also be managed. For example, Figure 3 shows how to produce a final product via co-operation between different items of equipment. In detail, to produce the final product ‘1’, feeds ‘1’ and ‘2’ are processed by NSE creating intermediate ‘a’. Feed ‘3’ is processed by CE to produce intermediate ‘b’. Then, intermediates ‘a’ and ‘b’ are further processed by SE to produce the final product ‘1’. Thus, to complete production, storage of intermediates plays an important role in terms of collaborations among different equipment items. Specifically, during manufacturing, if the production rate of early

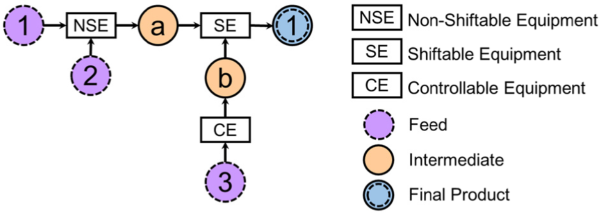


FIGURE 3. Co-operation between different equipment items.

equipment is higher than the utilization rate of later equipment, the resource must be temporarily stored in an intermediate storage unit. In contrast, if the production rate of early equipment is slower than the utilization rate of later equipment, the later equipment must go on standby until sufficient resources accumulate in the storage unit.

2) MATHEMATICAL REPRESENTATION

Based on the above architecture, we developed a series of equations quantizing problems associated with industrial DR, including a model of the energy demand, production resource balance, and objective functions. Note that, as our scheme is devised based on hourly prices announced by the utility company, the DR problem is discrete; the time unit for equipment operation is taken to be 1 hour. Thus, for ease of description, we divided the 24 hours of the day into 24 stages; each begins as an hour commences and expires when the next hour commences. For example, the first hour of the day, from 00:00 to 01:00, is termed stage 1, and stage 12 is the period from 11:00 to 12:00.

a: ENERGY DEMAND MODEL

Let i be the index of an equipment item and j the index of an OP. As any item of equipment can be in only a single state during 1 hour, Equation (1) constrains the OP choice of equipment, where λ_i^j is the binary indicator for OP $j(j \in [1, J])$ of equipment item $i(i \in [1, N])$. During stage t, if OP j is chosen, $\lambda_{i,t}^j$ is equal to 1. Otherwise, $\lambda_{i,t}^j$ is 0. On this basis, Equation (2) determines the energy demand of a single equipment item, where E_t^i denotes the energy demand of equipment item i at stage t, e_i^j denotes the corresponding energy demand of OP j, and J is the total number of OPs. The total energy demand (E_t^D) for all equipment items is summed using equation (3), where N is the total number of items of equipment.

$$1 = \sum\nolimits_{j=1}^{J} \lambda_{i,t}^j \tag{1}$$

$$E_t^i = \sum\nolimits_{j=1}^{J} \lambda_{i,t}^j e_i^j \tag{2}$$

$$E_t^D = \sum\nolimits_{i=1}^{N} E_t^i \tag{3}$$

b: PRODUCTION RESOURCE BALANCE MODEL

Let x be the index used to identify production resources. Equations (4) and (5) determine the total production for and the utilization of resource x (P_t^x and U_t^x respectively) at stage t, where $p_i^{j,x}$ is the production rate of resource x afforded by OP j of equipment item i, and $u_i^{j,x}$ is the utilization rate of resource x. Let R_t^x denote the residual of storage of resource x at the end of stage t. Then, the production resource balance is modeled by equation (6); the amount of a resource stored at the end of the current stage (R_t^x) depends only on the amount stored during the previous stage (R_{t-1}^x) and new production (P_t^x) and utilization (U_t^x) during the current stage t.

$$P_t^x = \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{J} p_i^{j,x} \lambda_{i,t}^j \tag{4}$$

$$U_t^x = \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{J} u_i^{j,x} \lambda_{i,t}^j \tag{5}$$

$$R_t^x = R_{t-1}^x + P_t^x - U_t^x \tag{6}$$

c: OBJECTIVE FUNCTIONS

As mentioned earlier, a real-world industrial facility seeks to achieve production targets using minimal energy cost. On this basis, the objective functions of industrial DR are summarized in equations (7) and (8). Equation (7) defines the need to minimize the daily energy cost (thus over 24 stages); E_t^D is the entire energy demand during stage t and HP_t is the hourly electricity price notified by the utility company. Equation (8) reflects the fact that production must not be compromised; the residual of final product storage (R_{24}^F) at the end of the last daily stage (stage 24) must thus be equal to or greater than the predefined production target.

$$min \left\{ \sum\nolimits_{t=1}^{24} E_t^D HP_t \right\} \tag{7}$$

$$R_{24}^F \geq Production\ target \tag{8}$$

B. MARKOV DECISION PROCESS FRAMEWORK

Based on the above description, we here present our MDP framework for industrial energy management. An MDP is a mathematical framework modeling decision-making in situations where outcomes are partly random and partly controllable, and has been widely adopted to solve optimization problems via RL [41]. In general, the standard MDP is a four-tuple (S, A, T, R) , where S and A denote the state and action, respectively, T is the state transition probability, and R is the reward function. However, as the state transition probability is not necessarily required for RL [18], [42], we here consider only the state, action, and reward function.

1) STATE FORMULATION

From the viewpoint of the FEMS, the composite state of the environment (S) includes the state within the facility (S^f) and the external state (S^e). In our case, the state of the industrial facility is defined as the residual of each production resource storage, because this can be used to follow production status in real time. The external state is defined as the real-time electricity price, which is important when seeking to minimize energy costs. Accordingly, equation (10) shows a sample of the state at stage $t(s_t)$, featuring a time indicator of the current stage (t), the current electricity price (HP_t) notified by the utility company, and the real-time storage residual of all resources $(R_t^1, R_t^2, \ldots, R_t^x)$.

$$S = S^f \cup S^e \tag{9}$$

$$s_t = (t, HP_{t,}R_t^1, R_t^2, \ldots, R_t^x) \tag{10}$$

2) ACTION FORMULATION

The FEMS schedules the energy demands of all energy loads in the facility. Thus, the learning agent is instructed to determine the OPs for all items of CE and SE. Equation (11) shows a sample action a_t taken at stage t, where a_t^i is the OP chosen for equipment item i at stage t, N is the total number of equipment items.

$$a_t = (a_t^1, a_t^2, \ldots, a_t^i, \ldots, a_t^N) \tag{11}$$

3) REWARD FUNCTION FORMULATION

In terms of the industrial DR described in Section II-A, this multi-objective optimization must both reduce energy costs and ensure that production is complete. Therefore, the reward function is formulated by (12), which simultaneously considers rewards for resource production (r_t^p) and energy cost (r_t^c).

$$r_t = r_t^p - r_t^c \tag{12}$$

$$r_t^p = \sum\nolimits_1^x P_t^x \tag{13}$$

$$r_t^c = E_t^D HP_t \tag{14}$$

In a real production line, feeds (raw materials) must be processed through several equipment items to create the final product. Thus, from the perspective of production resources, whenever materials are processed into a new type, they progress toward the final product. We define the production reward (r_t^p) in (13); the value depends on the all resource productions during the current stage. In detail, x is an index of different resources and P_t^x is the amount of resource x production during stage t. On this basis, as RL seeks to maximize the cumulative reward, the learning agent will schedule all equipment items to process all available resources (to earn the production reward). Thus, maximization of $\sum_{t=1}^{24} r_t^p$ implies that all available resources are processed into the final product within the 24 stages; production is complete. On the other hand, equation (14) defines another reward item r_t^c, which can be simply viewed as a punishment meted out for energy consumption. Specifically, E_t^D denotes the entire energy demand at stage t, during which time HP_t is the electricity price.

$$\max \sum_{t=1}^{24} r_t = \max \sum_{t=1}^{24} r_t^p - \min \sum_{t=1}^{24} r_t^c \tag{15}$$

In summary, the reward function of (12) both minimizes energy costs and ensures full production in an industrial DR environment; optimization attracts the highest cumulative rewards in all 24 stages. Based on this MDP framework, after the learning process converges, (15) is the maximized value; the term max $\sum_{t=1}^{24} r_t^p$ ensures completion of production and the term min $\sum_{t=1}^{24} r_t^c$ minimizes energy costs.

III. DEEP REINFORCEMENT LEARNING ALGORITHM

Based on our MDP framework for industrial energy management, we now present an actor-critic-based deep RL algorithm that we use to determine the optimal energy management policy for industrial facilities when real-time electricity prices vary. We first present the background on actor-critic and deep RL, and then our algorithm.

A. THE ACTOR-CRITIC

During interaction with the environment, the goal of the learning agent is to choose a sequence of actions maximizing the cumulative reward. On this basis, the function that indicates the action to choose in a certain state is termed a policy [43], denoted by $\pi(a|s)$. The cost-to-go function used to evaluate a state or a state-action pair is termed a value function. In detail, equation (16) formulates a state value function $V(s_t)$ used to estimate the expected cumulative reward from state $s_t.E$ denotes the expectation, r_t is the reward, and $0 < \gamma < 1$ is a discount factor. Thus, the state-action value function $Q(s_t, a_t)$ is used to estimate the expected cumulative reward if action a_t is selected in state s_t. RL algorithms can be divided into three types: value-based (critic-only), policy-based (actor-only), and actor-critic. The words actor and critic are synonyms of the policy and value function, respectively [10].

$$V(s_t) = E\left\{\sum\nolimits_{t=0}^{\infty} \gamma^t r_t \,|\, s = s_t\right\} \tag{16}$$

$$Q(s_t, a_t) = E\left\{\sum\nolimits_{t=0}^{\infty} \gamma^t r_t \,|\, s = s_t, a = a_t\right\} \tag{17}$$

In value-based methods such as Q-learning [44] and SARSA [45], the learning agent uses only the value function to choose the action, and there is no explicit function for the policy. Specifically, when choosing an action, the agent uses the value function to estimate the expected rewards of all candidate actions and then makes a decision. For example, a straightforward way is to select the greediest action (that with the highest reward). However, to ensure optimization, the agent must iterate all candidate actions for every state to determine the best solution, but this is time-consuming, especially when the state and action spaces are large. Therefore, application of value-based algorithms alone is not practical in the real world; the state and action spaces are always complex and sophisticated. On the contrary, policy-based algorithms do not use the value function. Rather, they directly parameterize policy, and update parameters using the policy gradient method [46]. Thus, unlike the step-by-step process required when using the value function, the agent directly performs sequential policy-specific actions, and adjusts the action choice by reference to the cumulative reward only after proceeding through an entire learning episode (from the initial to the terminal state). However, as bias from any single step accumulates throughout the entire episode, single-step gradient estimations exhibit high-level variance; learning is thus slow.

To overcome the disadvantages of these above two types of RL, they are combined into the actor-critic that can be considered as an advanced version of the policy-based method. The actor still determines the actions, and the critic uses its

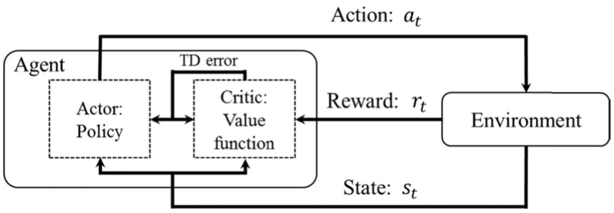


FIGURE 4. Schematic structure of actor-critic-based learning agent.

value function to evaluate current policy. Unlike what occurs when the actor works alone, joining of the critic reduces variance when estimating single-step gradients. Specifically, during policy optimization, the critic uses the value function to estimate the cumulative reward for all states that the agent has experienced. Then, differences between expected and received values are recorded as temporal difference (TD) errors, indicating whether the current policy is better or worse than expected. The actor uses the TD errors when updating action-choice gradients in each state, reducing variance and accelerating learning. Therefore, actor-critic algorithms usually exhibit better convergence speed than when an actor or critic works alone [14]. Figure 4 is the schematic of an actor-critic-based learning agent; the actor determines actions and the critic processes the rewards. During learning, after observing the latest state of the environment, the actor will choose the action in accordance with its current policy. On the other hand, the critic will evaluate the quality of this decision using the value function. The resulting TD error serves as feedback for both the actor and critic, based on which the policy and value function can be adjusted, respectively.

B. DEEP REINFORCEMENT LEARNING

During traditional RL using a simple lookup-table or linear function approximator, it is difficult to identify an optimal scheduling policy when the state space is large. This is termed the "curse of dimensionality" [47]. Thus, we exploited recent advances in the training of deep neural networks to handle such complexity [48]. Especially, AlphaGo [19] encouraged an intuitive understanding of what deep RL could achieve, and deep neural networks have proven to be powerful function approximator [39]. Here, as the MDP framework for industrial DR features multiple state inputs and multiple action outputs, we used the deep neural network to approximate both the policy and value function. Figure 5 shows an example implementation of the actor and critic networks. Specifically, in line with the general structure of the actor-critic architecture in Figure 4, the inputs to both networks are states of the agent's environment; this contains m-dimensional subitems numbered s_1 to s_m. Accordingly, the actor network output features n units labeled a_1 to a_n; these are the chosen actions. On the other hand, the output of the critic network is a state value v; this is the expected cumulative reward from the current input state. On this basis, the TD error (the difference between the estimated state value and the real reward) is the reinforcement signal used to adjust the weights of both the actor and critic networks.

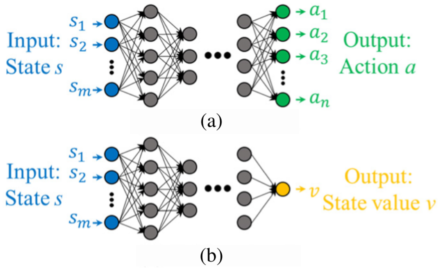


FIGURE 5. Example implementation of the actor and critic network.

C. DEEP RL ALGORITHM FOR INDUSTRIAL DR MANAGEMENT

Based on the above introduction, the entire pseudocode of the proposed algorithm is provided in Algorithm 1. As stated

Algorithm 1 Deep RL Algorithm for Industrial DR Management

```
Initialize the experience buffer for $s_t, a_t, r_t, s_{t+1}$
Initialize actor network $\pi\ (a_t \mid s_t, \theta)$ and critic network $V(s_t, \theta_v)$
Initialize episode counter $K = 1$
For episode $K = 1, 2, \ldots, K_{max}$ do:
    Initialize step counter $t = 1$
    Reset gradients $d\theta = 0$ and $d\theta_v = 0$
    Receive initial state $s_t$
    For step $t = 1, 2, \ldots, 24$ do:
        Perform $a_t$ according to policy $\pi\ (a_t \mid s_t, \theta)$
        Receive reward $r_t$ and new state $s_{t+1}$
        Store the sample $(s_t, a_t, r_t, s_{t+1})$ into the experience buffer
        $t = t + 1$
    End for
    For step $t = 24, \ldots 2, 1$ do:
        Get sample $(s_t, a_t, r_t, s_{t+1})$ from the experience buffer
        Calculate cumulative discounted reward:
            $R_t = \begin{cases} r_t + \gamma R_{t+1}, t < 24 \\ r_t, t = 24 \end{cases}$
        Estimate state value through the value function:
            $R_{es} = V(s_t, \theta_v)$
        Calculate temporal difference (TD) error:
            $\delta_t = R_t - R_{es}$
        Accumulate gradients for actor network:
            $d\theta = d\theta + \nabla_\theta log\pi\ (a_t \mid s_t, \theta)\, \delta_t + \beta \nabla_\theta H(\pi\ (a_t \mid s_t, \theta))$
        Accumulate gradients for critic network :
            $d\theta_v = d\theta_v + \partial {\delta_t}^2 / \partial \theta_v$
        $t = t - 1$
    End for
    Update the weights $\theta$ in actor network using $d\theta$ and the weights $\theta_v$ in critic network using $d\theta_v$
    Clear the experience buffer
    $K = K + 1$
End for
```

above, this is a variant of the actor-critic algorithm; both the actor and the critic are estimated by the deep neural network. Thus, we use $\pi(a \mid s, \theta)$ to denote the actor network and $V(s, \theta_v)$ to denote the critic network, where θ and θ_v are parameters within the two networks, respectively. The parameters are updated using the policy gradient method [47].

From the macro view, the algorithm can be decomposed into three stages: initialization (lines 0 to 2), accumulation of experience (lines 7 to 12), and learning from experience (lines 13 to 21). In addition, the entire learning process is controlled by two variables: the episode counter K (line 3) and the step counter t (lines 7 and 13). In the context of hourly price-based DR, we define a learning episode as a day, and a step as an hour. Thus, a learning episode features 24 steps.

During initialization (line 0), the buffers for the state (s_t), action (a_t), reward (r_t), and next state (s_{t+1}) are initialized, to allow the experience that will be accumulated to be stored for future learning. Then, the neural networks for the actor and critic are initialized using the random parameters θ and θ_v, respectively, in line 1. Commencing on line 3, the algorithm enters episodic iteration; K_{max} is the number of iterations required for the convergence that determines an optimal energy management policy. At the beginning of each episode, i.e., from lines 4 to 6, the agent resets the step counter $t = 1$, and the gradients of both the policy and the value function network parameters are reset to 0. Then, the agent begins to observe the environment's initial state s_t.

Commencing on line 7 and proceeding to line 12, the algorithm engages in experience accumulation. In detail, as the step counter t increases, the agent sequentially chooses action a_t based on state s_t by reference to the policy $\pi\ (a_t \mid s_t, \theta)$, during this process, each pair of samples (s_t, a_t, r_t, s_{t1}) is stored in the experience buffer to allow for future learning.

After an entire episode (24 steps) has been executed, from lines 13 to 21, the algorithm enters the learning-from-experience phase. In detail, line 14 reads sample (s_t, a_t, r_t, s_{t1}) from the experience buffer and, based on these, sums the discounted rewards for each state in line 15, where $0 < \gamma < 1$ is the discount factor. In line 16, the estimated value for each state is generated by reference to the current value function. Then, line 17 calculates the TD error δ. Here, a positive δ means that the current policy is 'good' because it created a state with a better-than-expected reward. On the contrary, a negative δ means that the current policy is 'bad' because it created a state with a worse-than-expected reward.

Based on the TD errors, the actor and critic accumulate their gradients in lines 18 and 19, respectively, using the general method of [46]. Moreover, using the method of [49], an entropy regularization term $\beta \nabla_\theta H(\pi(a_t|s_t, \theta))$ is added to actor policy gradient to encourage exploration, where H is entropy and β is the exploration factor. Then, the parameters of both the policy and value function networks are updated in line 22. Line 23 clears the experience buffer to allow for future storage of new experiences. Line 24 updates the episode counter. The learning process begins afresh until the cumulative reward reaches its maximum value.

IV. CASE STUDY AND SCHEME IMPLEMENTATION

To verify the feasibility of our DR scheme, we now conduct a real-world case study of SPM. First, we introduce the detailed manufacturing process and then the implementation of our scheme.

A. STEEL POWDER MANUFACTURING

The field of powder metallurgy creates opportunities that are lacking when materials are in their conventional forms; melting is not required during complex component formation [50]. Thus, metallic powders for additive manufacturing (three-dimensional printing) are widely used in, for example, the aerospace, medical, and rapid tooling fields. Figure 6 [51] provides an overview of SPM, which is linear and associated with strict specifications for all processing steps. In detail, the process features an atomizer, a dehydrator, a dryer, two crushers, two classifiers, a magnetic separator, a reduction furnace, and a blender. As the principal intermediate products differ, we divide the process into three phases. The first is iron powder fabrication. Obviously, the atomizer is a key component of the entire process, transforming molten iron into iron powder via several high-pressure water jets. Next, the powder slurry is dewatered by the dehydrator and dryer. Steel powder is formed in the second step. As the powder particles may differ in size, the dry powder must be further homogenized in the crusher and classifier. The magnetic separator is used to remove any remaining mixed slag. Then, the iron powder is deoxidized in the reduction furnace and changed into steel. In the third and last step, as powder condensation is common, attributable to the high temperature of the reduction furnace, the steel powder must be further refined by the crusher and classifier to control particle diameter. Then, the blender is used to mix other materials with the powder to satisfy market demands.

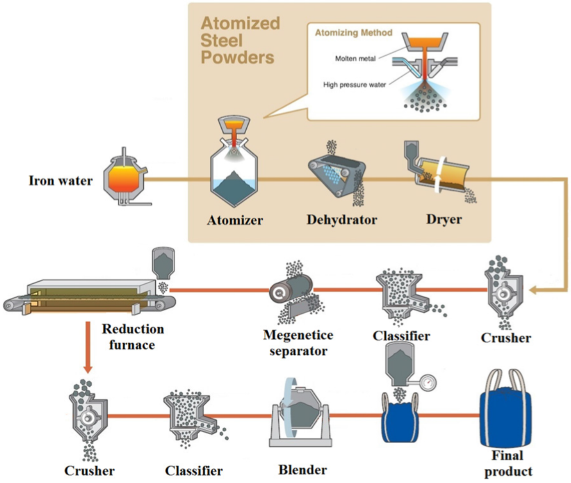


FIGURE 6. Steel powder manufacturing process.

To mirror the real situation in the factory, apart from the principal production line equipment, we add two essential auxiliary components. One is a water-cooling tower (WCT), which generates the cool water indispensable in terms of metal atomization, and cooling of the dryer and reduction furnace. The other is a nitrogen generator (NG) delivering pure nitrogen to the reduction furnace, ensuring an oxygen-deprived atmosphere.

B. SCHEME IMPLEMENTATION

Tables 1–5 show the detailed SPM parameters. Specifically, Table 1 lists the energy demands of all equipment items, including the atomizer, dehydrator, dryer, crusher, classifier, magnetic separator, reduction furnace, blender, WCT, and NG. Tables 2–4 list their resource-processing capacities. Table 5 details the relationship between each equipment item and resource storage. The production target was set to 120 tons of steel powder per day. Note that, apart from the number of OPs, during simulation, the learning agent does not access any detailed information from the SPM process, such as the equipment's resource-processing capabilities, energy demand, or the relationships between equipment items and production resources. The MDP framework solving the SPM energy management problem is as follows:

TABLE 1. Detailed information on energy demand.

Equipment name	Equipment type	Energy demand (kWh)				
		OP #1	OP #2	OP #3	OP #4	OP #5
Atomizer	SE	0	60	/	/	/
Dehydrator	SE	0	10	/	/	/
Dryer	SE	0	30	/	/	/
Crusher	CE	0	15	20	/	/
Classifier	CE	0	15	25	/	/
Magnetic separator	SE	0	10	/	/	/
Reduction furnace	NSE	75	/	/	/	/
Blender	CE	0	6	10	/	/
Water cooling tower	CE	0	25	50	/	/
Nitrogen generator	CE	0	20	30	40	55

TABLE 2. Processing capability of equipment in the main process.

Equipment name	Processing rate (ton/h)		
	OP #1	OP #2	OP #3
Atomizer	0	30	/
Dehydrator	0	20	/
Dryer	0	15	/
Crusher	0	10	15
Classifier	0	10	20
Magnetic separator	0	30	/
Reduction furnace	15	/	/
Blender	0	10	15

TABLE 3. Cool water processing capability.

Equipment name	Processing rate (m^3/h)		
	OP #1	OP #2	OP #3
Water cooling tower	0	100	200
Atomizer	0	50	/
Dryer	0	10	/
Reduction furnace	20	/	/

TABLE 4. Pure nitrogen processing capability.

Equipment name	Processing rate (Nm^3/h)				
	OP #1	OP #2	OP #3	OP #4	OP #5
Nitrogen generator	0	40	60	80	100
Reduction furnace	20	/	/	/	/

1) STATE FORMULATION

In line with the MDP framework of Section II-B, at stage t, as shown in (18), the environment consists of an indicator of hourly stage (t), the day-ahead hourly electricity price ($HP_{t,}$), and the real-time residuals of the 13 resources listed in Table 5.

$$s_t = (t, HP_t, R_t^{molten\ iron}, \ldots, R_t^{steel\ powder}, R_t^{cool\ water}, R_t^{pure\ nitrogen}) \quad (18)$$

TABLE 5. Detailed information on each resource storage.

Resource name	Inlet equipment	Outlet equipment
Molten iron	/	Atomizer
Powder slurry	Atomizer	Dehydrator
Wet powder	Dehydrator	Dryer
Dry powder	Dryer	Crusher #1
Crude powder	Crusher #1	Classifier #1
Impure powder	Classifier #1	Magnetic separator
Semi-finished powder	Magnetic separator	Reduction furnace
Annealed powder	Reduction furnace	Crusher #2
Condensed powder	Crusher #2	Classifier #2
Pure powder	Classifier #2	Blender
Steel powder	Blender	/
Cool water	Water cooling Tower	Atomizer Dryer Reduction furnace
Pure nitrogen	Nitrogen generator	Reduction furnace

2) ACTION FORMULATION

As shown in (19), the action taken at each stage involves determination of the OPs for all 10 equipment items listed in Table 1.

$$a_t = (a_t^{atomizer}, \ldots, a_t^{blender}, a_t^{water\ cooling\ tower}, a_t^{nitrogen\ generator}) \quad (19)$$

3) REWARD FUNCTION FORMULATION

The reward is calculated using (12), (20), and (21), where r_t^p is the sum of resource production during stage t and r_t^c is the total energy cost for each equipment item; P denotes the amount of resource production, E is the energy demand, and HP_t is the electricity price.

$$\begin{aligned} r_t^p &= P_t^{powder\ slurry} + P_t^{wet\ powder} + \ldots + P_t^{steel\ power} \\ &\quad + P_t^{cool\ water} + P_t^{pure\ nitrogen} \quad (20) \\ r_t^c &= HP_t\big(E_t^{atomizer} + \ldots + E_t^{blender} + E_t^{water\ cooling\ tower} \\ &\quad + E_t^{nitrogen\ generator}\big) \quad (21) \end{aligned}$$

TABLE 6. Determined operating point of each CE and SE.

Hourly stages	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Atomizer	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Dehydrator	1	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Dryer	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Crusher #1	1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1
Classifier #1	1	1	1	1	3	3	3	3	3	3	2	3	1	1	1	1	1	1	1	1	1	1	1	1
Magnetic separator	1	1	1	1	1	2	2	2	2	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1
Crusher #2	1	1	1	1	1	1	1	3	3	3	3	3	3	1	1	1	1	1	1	3	3	1	1	1
Classifier #2	1	1	1	1	1	1	1	1	1	3	3	3	1	3	1	1	1	1	1	1	3	3	1	1
Blender	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	1	1	1	1	1	1	3	3	3
Water cooling tower	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Nitrogen generator	4	5	4	4	5	4	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

V. RESULTS AND ANALYSIS

Applying the algorithm of Section III to the case study of Section IV, the learning agent and the corresponding SPM process are programmed with the aid of TensorFlow [52], an open-source software library used to develop AI-related applications. The day-ahead hourly electricity prices used in the simulation were obtained from ComEd, a utility company operating in the Pennsylvania-New Jersey-Maryland (PJM) electricity market [53].

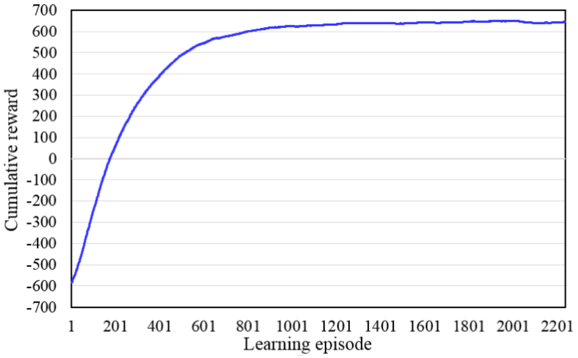


FIGURE 7. Cumulative reward during the learning process.

To demonstrate the performance of the proposed scheme, the detailed simulation results based on the electricity price for July 20, 2017 are discussed in detail in this section. Figure 7 shows the cumulative reward as the number of training episodes increased. Initially, the reward is relatively low because the learning agent is engaging in trial-and-error. As experience is gained via episodic iteration, the learning agent constantly adjusts its policy and the cumulative reward gradually increases. Finally, the algorithm converges at about episode 1,800, yielding the optimal policy. Table 6 shows the selected actions of that optimal policy, thus the OPs of CE and SE during each stage. To better illustrate all of these actions, Figure 8 shows the corresponding aggregated energy demands. As expected, our scheme exhibits the desired DR behavior, shifting energy demand from high-price stages to low-price stages. Taking the WCT and the NG as examples, when the price is low, such as during stages 1–7, both were scheduled to work at high energy-demand OPs (Table 6); considerable energy was consumed. Correspondingly, nitrogen and cool water were stored, as shown in Figure 9. In contrast, when the price was high, such as during stages 16–20, both equipment items worked at low energy-demand OPs (Table 6); energy demand decreased, as did nitrogen and cool water storage (Figure 9). Thus, stored resources were produced during periods when electricity was cheaper and became exhausted during periods of high-price electricity.

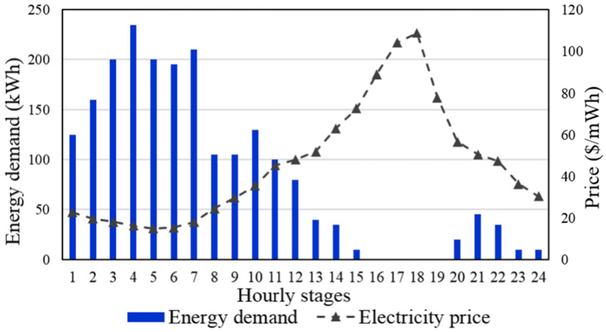


FIGURE 8. Aggregated energy demand of the CE and SE.

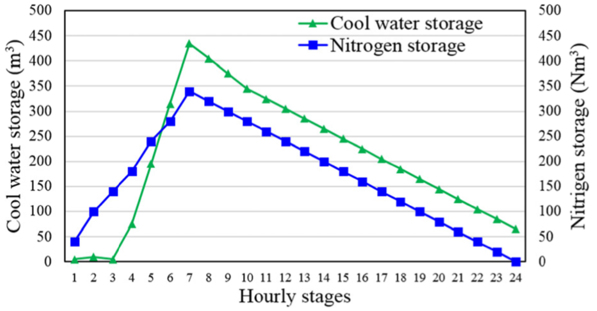


FIGURE 9. Residual in the storage for cool water and pure nitrogen.

To further emphasize the capabilities of our DR scheme, Figure 10 compares the energy demands of the entire manufacturing process during each stage. Specifically, Case 1 (brown) shows the demand when electricity price fluctuations were ignored; all equipment items simply operated in sequence to complete production; no DR algorithm was involved. Case 2 (blue) shows demand based on our DR scheme. The gray dashed line with the triangles represents the electricity price. The energy demand during the relatively

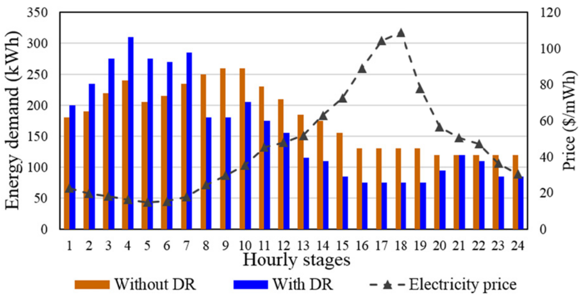


FIGURE 10. Total energy demand comparison.

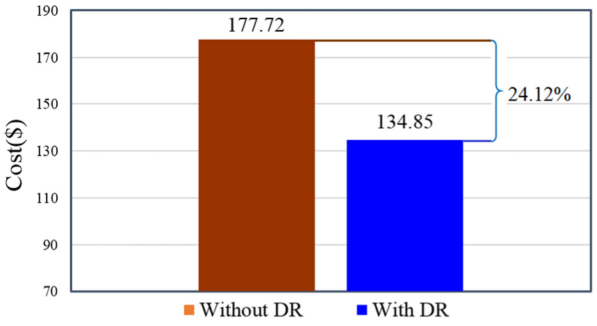


FIGURE 11. Total expense comparison.

high-price period was reduced by our DR scheme compared to that of Case 1, indicating that our scheme affords efficient DR control, eliminating peak loads. Figure 11 compares the total expenses. The total energy cost of Case 2 (blue) using our DR scheme was 24.12% less than that of Case 1 (brown). Thus, our DR scheme reduces energy costs.

VI. CONCLUSION AND FUTURE WORK

We present a deep RL-based industrial DR scheme optimizing industrial energy management. To ensure practical application, we designed an MDP framework for industrial DR and formulated corresponding state, action, and reward function. We used an actor-critic-based deep RL algorithm to determine the most efficient manufacturing schedule; both the policy and value function were estimated by the deep neural network. We simulated a real-world manufacturing process and our DR scheme balanced energy demand, reducing energy costs and ensuring that production targets were met.

In the future, we will consider applying RL to a wider range of DR applications. Compared to the large number of studies on price-based DR, incentive-based DR is indispensable but there has been much less research in this area. We will attempt to build a suitable RL framework for incentive-based DR applications.

REFERENCES

[1] (2018). Annual Energy Outlook. US Energy Information Administration (EIA), Washington, DC, USA, Accessed: Feb. 2018. [Online]. Available: https://www.eia.gov/outlooks/aeo/pdf/AEO2018.pdf

[2] (2017). Tracking Clean Energy Progress. IEA Publications, Paris, France. [Online]. Available: https://www.iea.org/publications/freepublications/publication/TrackingCleanEnergyProgress2017.pdf

[3] G. N. Paterakis, O. Erdinç, and J. P. S. Catalão, "An overview of demand response: Key-elements and international experience," *Renew. Sustain. Energy Rev.*, vol. 69, pp. 871–891, Mar. 2017.

[4] T. Samad, E. Koch, and P. Stluka, "Automated demand response for smart buildings and microgrids: The state of the practice and research challenges," *Proc. IEEE*, vol. 104, no. 4, pp. 726–744, Apr. 2016.

[5] M. Matteo and G. Rizzoni, "Residential demand response: Dynamic energy management and time-varying electricity pricing," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1108–1117, Mar. 2016.

[6] M. L. Tuballa and M. L. Abundo, "A review of the development of smart grid technologies," *Renew. Sustain. Energy Rev.*, vol. 59, pp. 710–725, Jun. 2016.

[7] M. H. Shoreh, P. Siano, M. Shafie-Khah, V. Loia, and J. P. S. Catalão, "A survey of industrial applications of demand response," *Electr. Power Syst. Res.*, vol. 141, pp. 31–49, Dec. 2016.

[8] A. Aghajanzadeh, A. McKane, C. Wray, and P. Therkelsen, "2006–2015 research summary of demand response potential in california industry, agriculture and water sectors," Ernest Orlando Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-1004131, Aug. 2015, pp. 1–66.

[9] A. T. McKane, "Opportunities, barriers and actions for industrial demand response in California," Ernest Orlando Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-1335E, Jan. 2008, pp. 1–89.

[10] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[11] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*. [Online]. Available: https://arxiv.org/abs/1701.07274

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[14] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: A comparison on a power system problem," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 517–529, Apr. 2009.

[15] L. Roveda, G. Pallucca, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Iterative learning procedure with reinforcement for high-accuracy force tracking in robotized tasks," *IEEE Trans. Ind. Inform.*, vol. 14, no. 4, pp. 1753–1763, Apr. 2018.

[16] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 653–664, Mar. 2017.

[17] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2018.

[18] T. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[19] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[20] Q. Wei, D. Liu, and H. Lin, "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 840–853, Mar. 2016.

[21] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Appl. Energy*, vol. 235, pp. 1072–1089, Feb. 2019.

[22] K. Bruninx, D. Patteeuw, E. Delarue, L. Helsen, and W. D'Haeseleer, "Short-term demand response of flexible electric heating systems: The need for integrated simulations," in *Proc. Int. Conf. Eur. Energy Market (EEM)*, 2013, pp. 1–10.

[23] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuška, and R. Belmans, "Reinforcement learning applied to an electric water heater: From theory to practice," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3792–3800, Jul. 2016.

[24] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuska, and R. Belmans, "Residential demand response of thermostatically controlled loads using batch reinforcement learning," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2149–2159, Sep. 2017.

[25] F. Ruelens, B. J. Claessens, S. Vandael, S. Iacovella, P. Vingerhoets, and R. Belmans, "Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning," in *Proc. 18th IEEE Power Syst. Comput. Conf. (PSCC)*, Wroclaw, Poland, Aug. 2014, pp. 1–7.

[26] O. De Somer, A. Soares, K. Vanthournout, F. Spiessens, T. Kuijpers, and K. Vossen, "Using reinforcement learning for demand response of domestic hot water buffers: A real-life demonstration," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Turin, Italy, Sep. 2017, pp. 1–7.

[27] C. Patyn, F. Ruelens, and G. Deconinck, "Comparing neural architectures for demand response through model-free reinforcement learning for heat pump control," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Limassol, Cyprus, Jun. 2018, pp. 1–6.

[28] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, to be published.

[29] A. Chiş, J. Lundén, and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3674–3684, May 2017.

[30] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, Jul. 2015.

[31] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 409–414.

[32] Z. Wen, D. O'Neill, and H. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sep. 2015.

[33] A. T. Kaliappan, S. Sathiakumar, and N. Parameswaran, "Flexible power consumption management using Q learning techniques in a smart home," in *Proc. IEEE Conf. Clean Energy Technol. (CEAT)*, Nov. 2013, pp. 342–347.

[34] Y. Liu, C. Yuen, N. U. Hassan, S. Huang, R. Yu, and S. Xie, "Electricity cost minimization for a microgrid with distributed energy resource under different information availability," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2571–2583, Apr. 2015.

[35] X. Qiu, T. A. Nguyen, and M. L. Crow, "Heterogeneous energy storage optimization for microgrids," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1453–1461, May 2016.

[36] P. Kofinas, A. I. Dounis, and G. A. Vouros, "Fuzzy Q-learning for multi-agent decentralized energy management in microgrids," *Appl. Energy*, vol. 219, pp. 53–67, Jun. 2018.

[37] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[38] G. Wen, C. L. P. Chen, J. Feng, and N. Zhou, "Optimized multi-agent formation control based on an identifier–actor–critic reinforcement learning algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2719–2731, Oct. 2018.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[40] *Industrial-Process Measurement, Control and Automation System Interface Between Industrial Facilities and the Smart Grid*, document IEC TS 62872 ED1, 2015.

[41] M. H. A. Davis, *Markov Models and Optimization*. Evanston, IL, USA: Routledge, 2018.

[42] H. Y. Ong, K. Chavez, and A. Hong, "Distributed deep Q-learning," 2015, *arXiv:1508.04186*. [Online]. Available: https://arxiv.org/abs/1508.04186

[43] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[44] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[45] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Univ. Cambridge, Cambridge, U.K., Tech. Rep. CUED/FINFENG/ TR 166, 1994.

[46] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2000, pp. 1057–1063.

[47] F. Bach, "Breaking the curse of dimensionality with convex neural networks," *J. Mach. Learn. Res.*, vol. 18, no. 19, pp. 629–681, 2017.

[48] R. F. Atallah, C. M. Assi, and M. J. Khabbaz, "Scheduling the operation of a connected vehicular network using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1669–1682, May 2019.

[49] R. J. Williams and J. Peng, "Function optimization using connectionist reinforcement learning algorithms," *Connection Sci.*, vol. 3, no. 3, pp. 241–268, 1991.

[50] J. M. C. Azevedo, A. CabreraSerrenho, and J. M. Allwood, "Energy and material efficiency of steel powder metallurgy," *Powder Technol.*, vol. 328, pp. 329–336, Apr. 2018.

[51] *Reduced Iron Powders, Atomized Iron and Steel Powders*. Accessed: Jun. 25, 2019. [Online] Available: https://www.jfe-steel.co.jp/en/products/ironpowders/catalog/j1e-001.pdf

[52] *TensorFlow*. Accessed: Jun. 25, 2019. [Online]. Available: https://www.tensorflow.org/

[53] PJM Interconnection LLC. *Locational Marginal Prices*. Accessed: Jun. 25, 2019. [Online]. Available: http://www.pjm.com/markets-and-operations/energy/real-time/monthlylmp.aspx

XUEFEI HUANG received the B.S. degree in electronic and information engineering from the Wuhan University of Science and Technology, Wuhan, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Electronic Systems Engineering, Hanyang University, Ansan, South Korea.

His research interests include artificial intelligence, demand response in smart grid, smart manufacturing, and the Internet of Things.

SEUNG HO HONG (M'89–SM'10) received the B.S. degree from Yonsei University, Seoul, South Korea, in 1982, the M.S. degree from Texas Tech University, Lubbock, TX, USA, in 1985, and the Ph.D. degree from Pennsylvania State University, University Park, PA, USA, in 1989, all in mechanical engineering.

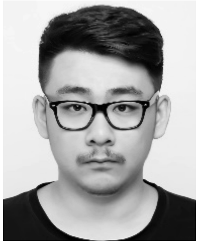
He was the Director of the Ubiquitous Sensor Networks Research Center, Hanyang University, a subsidiary of the Gyeonggi Regional Research Center Program. He was a Visiting Scholar with the National Institute of Standards and Technology, USA; the Vienna University of Technology, Austria; and Zhejiang University, China. He is currently a Professor with the Department of Electronic Engineering and also the Director of the Connected Smart Systems Laboratory, Hanyang University, Ansan, South Korea. He is also a Visiting Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences; the Chongqing University of Posts and Telecommunications; and the Wuhan University of Science and Technology, China. He is also a Foreign Expert with the Tianjin University of Technology through the Tianjin Thousand Talents Program. His research interests include smart manufacturing, smart grid, the industrial IoT, cyber-physical systems, and artificial intelligence.

MENGMENG YU (M'16) received the B.S. degree in communication engineering from the Wuhan University of Technology, Wuhan, China, in 2008, the M.S. degree in detection technique and automation equipment from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2011, and the Ph.D. degree in electronic systems engineering from Hanyang University, Ansan, South Korea, in 2015.

She was a Postdoctoral Researcher under the BK21 PLUS Program (BK21+) with Hanyang University, from 2015 to 2018. She is currently a Research Professor with Hanyang University. Her research interests include smart grid, game theory, machine learning, and smart manufacturing. She received the Outstanding Researcher Award of BK21+ and the Best Paper Award from the Workshop on Smart City Infrastructure and Applications, in 2016.

YUEMIN DING (M'18) received the Ph.D. degree in electronic systems engineering from Hanyang University, Ansan, South Korea, in 2014.

He is currently an Associate Professor with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. His research interests include the Internet-of-Things, smart grid communications, smart homes/buildings, and complex networks.



JUNHUI JIANG received the B.S. degree in electronic information engineering from the Harbin University of Science and Technology, Harbin, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electronic Systems Engineering, Hanyang University, Ansan, South Korea.

His research interests include time-sensitive networking and smart manufacturing.